Drifting and Popularity: A Study of Time Series Analysis of Topics

Muhammad Haseeb UR Rehman Khan
University of Tsukuba
Tsukuba, Japan
s2036048@s.tsukuba.ac.jp

Kei Wakabayashi
University of Tsukuba
Tsukuba, Japan
kwakaba@slis.tsukuba.ac.jp

***Abstract*—Topic modeling is extensively used for the Natural Language Processing (NLP) problems of summarizing, organizing, and understanding large document datasets. Latent Dirichlet Allocation (LDA) is widely used for the collection of topics, whereas Dynamic Topic Model (DTM) is famous for the time-series topic analysis. However, by estimating the number of occurrences of topics in each time slice, we can obtain time-series topic popularity using standard LDA. Therefore, if this can be extracted with LDA, then why do we need DTM which has a very high computation cost? The purpose of this research is to determine, either time-series topic information can be extracted from LDA or we need DTM. Topic drifting and popularity are two fundamental aspects of time-series topic analysis. We conducted experiments with multiple datasets to check the reliability of the information extracted from both models. We used Jensen-Shannon (JS) similarity-based analysis to check for information overlap. We constructed time-series topic popularity graphs for both models from the document-topic distributions and compared the results. Our results show that there is notable DTM topic drifting information in some cases and sometimes no or vague topic drifting. Topic drifting embedded in DTM topics makes this model less favorable for topic popularity analysis. On the other hand, LDA topics with no time transition information provided concrete results of topic popularity.**

***Keywords*—DTM; LDA; Topic Modeling; Time Series Analysis.**

I. INTRODUCTION

Latent Dirichlet Allocation (LDA) [1] and Dynamic Topic Model (DTM) [2] are widely used topic models that revolutionized the solving of topic modeling-based NLP problems. Situations that need the assistance of topic models often involve time-series document collections, including Twitter posts, news articles, and academic paper archives. By focusing on the nature of time-series, many useful applications can be developed, such as bursty topic detection [3], trend analysis [4][5], topic evolution analysis [6][7][8], topic transition pattern mining [9], etc.

To capture the time-series features of topics, DTM and its related-models [8][10] assume dynamic drift of distributions. Although the DTM-based models appropriately find topics over time, they require expensive computational cost, which can be a critical drawback in some applications. On the other hand, there is a large body of work developing efficient inference algorithms for LDA [11][12][13] because of its simpler architecture compared to DTM. While both models learn and work differently and even give different results, some practitioners and researchers employ LDA instead to analyze the time-series nature to take advantage of its efficiency, and these attempts seem to be successful according to the literature [14].

The question that arises in this background is; if time series topics information can be extracted by using LDA, which is faster than DTM, then why do we need to use DTM? To answer this question this research is conducted with a problem statement "*Can time-series topic information of DTM be extracted from LDA?*" To the best of our knowledge, there have been no studies that extensively compared the information extracted using LDA with that of DTM.

Topic drifting and topic popularity are fundamental time-series information that can be extracted from DTM. Topic drifting is the topic transition over time and popularity is the measure of topic proportion at each time slice. The challenging part in topic transition analysis is that, DTM topic set has a sequential structure whereas LDA topic set has no sequential information at all. To map the unstructured topic set with DTM topics, we used a probability distribution similarity method.

Based on this matching, we analyzed both topic sets, and in this process, we encountered with fragmentation issue, which we will describe later (Figure 1). DTM provides the time evaluation of topics, which means one single DTM topic can shift to a new subject if compared with the initial time's topic subject, whereas LDA topic's theme remains the same because LDA has no time aspect. This shifting in DTM topics is called fragmentation. In this experiment, we found that some DTM topics contain the information of two or more LDA topics; in other words, they have two or more fragmented topics.

We built time-series topic population graphs for topic popularity analysis. There are pros and cons for each model. LDA extracts the focus on the collection of topics, whereas DTM can find connections between different themes and how subjects interchange within the same domain or topic.

Even though DTM has the edge of finding topic transitions over time, mostly constructing only population graphs for LDA topics is enough for time-series analysis [14]. Some specific problems require DTM to extract topic transition despite its high computation cost [15].

The rest of the paper is organized as follows: In the next section, we describe the closely related background research. In Section 3, we present an overview of computing time series topic estimation for LDA topics. Section 4 describe about similarity analysis. Section 5 is about datasets and models used for the experiments. Results are shown in Section 6. At the end, few discussion points are mentioned in Section 7 and conclusion is presented in the last section.

 ISBN: 978-1-61208-842-6

II. Background

D. Koike et al. [3] proposed a method that draws a time-series graph to find the bursty topic detection in Twitter data individually, as well as with correlated news, by using DTM [2]. They extracted 50 topics from a subset of news articles and Twitter about *The London Olympics*. Khan et al. [14] performed a similar type of experiment using LDA. In their experiment, LDA was trained on 1000 topics on hashtag-pooled documents of English tweets. They then created an inference dataset from the same dataset using day-hashtag tweet pooling. In the end, they created time-series graphs of topics that showed the topic popularity, topic burstiness, and interval of bursty topics. In general, [3] and [14] are the same but used different topic models. We want to know why.

Before applying topic modeling, some preprocessing steps are required because documents are messy in general. Applying linguistic preprocessing may be of some help [16]. For Twitter dataset, tweet pooling was used that has been proposed as an intuitive solution [17][18] when models perform poorly because of small document size. Hashtag pooling [19] and day-hashtag [14] were used.

III. Time-Series Topic Estimation by LDA

LDA topics information is organized by time to compare it with DTM topics. LDA assumes a latent topic distribution for each document d denoted by θ_d and a latent topic assignment z_i for each word w_i in a document. The word w_i is drawn from a distribution of words associated to the assigned topic $z_i = k$, which is denoted by ϕ_k. We trained the LDA with multiple datasets without any modification to the LDA machinery. Formally, when we denote a set of documents that we would like to analyze by $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_D\}$, we simply use X as a training dataset for ordinary LDA training. Before the LDA training, we apply a pooling method when we deal with a short text dataset such as Twitter. In that case, each document $\mathbf{x}_d$ consists of multiple text instances (e.g., tweets). We denote the number of instances that are contained in a document $\mathbf{x}_d$ by T_d. If no pooling method is applied, $T_d = 1$ for all documents.

For the inference part that estimates the number of documents for each topic, we take the time information into account. Each document $\mathbf{x}_d$ is associated to a specific time slice, which we denote by $\tau(\mathbf{x}_d)$. Let $X_t = \{\mathbf{x} \in X | \tau(\mathbf{x}) = t\}$, be a set of documents in time slice t. We estimate the topic distribution θ_d for each document in X and calculate the estimated number of documents for each topic k at each time slice t, denoted by N_k^t, using the following equations:

$$N_k^t = \sum_{d:\mathbf{x}_d \in X_t} \theta_{dk} T_d \tag{1}$$

Probability distribution θ is calculated using Dirichlet distribution by applying LDA to the input data.

Given words $\mathbf{x} = w_1, \ldots, w_M$, we estimate the distribution of θ.

$$p(\theta|\mathbf{x}) = \sum_{\mathbf{z}} p(\theta|z) p(\mathbf{z}|\mathbf{x}) \tag{2}$$

where corresponding topics $\mathbf{z} = z_1, \ldots, z_K$, and the summation are over all possible assignments of $\mathbf{z}$. Since summation is analytically intractable, we apply Monte Carlo approximation with only one sample. We obtain a sample $\hat{z}$ from $p(z|x)$ using (collapsed) Gibbs sampling with five iterations. This approximation reduces the equation into the posterior probability of θ given $\hat{z}$.

$$p(\theta|x) \approx p(\theta|\hat{z}) \tag{3}$$

The posterior is a Dirichlet distribution of which the expectation $\hat{\theta}$ is:

$$\hat{\theta_k} = \frac{n_k + \alpha_k}{N + \sum_{k'} \alpha_{k'}} \tag{4}$$

where $n_k = \sum_{i=1}^{N} \delta(\hat{z}_i, k)$, i.e., the number of topic k in $\hat{z}$.

The final step is to estimate the number of documents to make the time-series popularity graphs.

IV. Similarity Analysis of DTM and LDA Topics

The DTM and LDA topics are in the form of word and probability distributions. We extract the top 50 words for all topics so word distribution is $K \times 50$ in LDA and K is the number of topics. Due to very low probability density of lower ranked words, Top 50 words are enough to convey the meaning of a topic. The top words may change in a DTM topic over time, so overall word distribution of a DTM topic varies, but it is $K \times 50$ for one time slice, the same as a LDA topic. To check the relation between topics, we use a widely accepted similarity measure, the Jensen-Shannon (JS) divergence [20].

We apply normalization on both the DTM and LDA topic-word distributions because we consider only top 50 words. We denote the normalized distribution for the kth LDA topic by $\tilde{\phi}_k$ and jth DTM topic at time slice t by $\tilde{\phi}_j^t$. The JS divergence between these distributions is defined as:

$$JSD(\tilde{\phi}_k || \tilde{\phi}_j^t) = \frac{1}{2} D(\tilde{\phi}_k || T_M) + \frac{1}{2} D(\tilde{\phi}_j^t || T_M) \tag{5}$$

where

$$T_M = \frac{1}{2}(\tilde{\phi}_k + \tilde{\phi}_j^t) \tag{6}$$

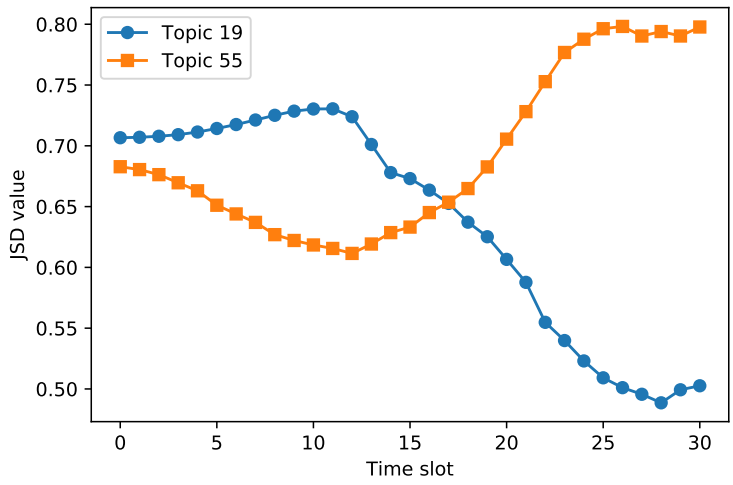
$D(\tilde{\phi}_k || \tilde{\phi}_j^t)$, is the Kullback-Leibler divergence:

$$D_{KL}(\tilde{\phi}_k || \tilde{\phi}_j^t) = \sum_{w \in W} P(w|\tilde{\phi}_k) \log \left(\frac{P(w|\tilde{\phi}_k)}{P(w|\tilde{\phi}_j^t)} \right) \tag{7}$$

A. Matching DTM and LDA topics

JS analysis tells us about the information overlap between DTM and LDA topics and is a good way to confirm either the topics are similar in both models or not. This analysis also illustrates fragmentation. An example is shown in Figure 1, where we see that the sample DTM topic was "Tensor decomposition for signal processing" at start, but later the topic's theme shifted rapidly towards "Tensor decomposition" and "Signal processing" was no longer significant. Whereas "Tensor decomposition" and "Signal Processing" are two different topics in LDA analysis. This phenomenon is called

T=0, (year = 1987)	T=5, (year = 1992)	T=10, (year = 1997)	T=15, (year = 2002)	T=20, (year = 2007)	T=25, (year = 2012)	T=30, (year = 2017)
matrix, vectors, components, component, analysis, principal, signals, signal, matrices, spectral, column, eigenvalues, source, orthogonal, eigenvectors, ...	matrix, component, components, analysis, principal, vectors, signal, source, signals, matrices, pca, sources, spectral, independent, separation, ...	matrix, component, components, analysis, independent, source, signal, sources, separation, principal, ica, signals, pca, blind, matrices, ...	matrix, signal, source, components, analysis, component, sources, signals, ica, matrices, independent, pca, separation, principal, subspace, ...	matrix, matrices, pca, rank, analysis, signal, components, source, columns, component, sources, reconstruction, subspace, ica, vectors, ...	matrix, rank, matrices, norm, low, pca, subspace, tensor, columns, entries, column, principal, singular, decomposition, completion, ...	matrix, rank, matrices, tensor, low, spectral, norm, subspace, decomposition, entries, error, singular, sparse, completion, pca, ...



LDA Topic 19	LDA Topic 55
rank, matrices, norm, tensor, entries, decomposition, columns, column, subspace, spectral, singular, completion, privacy, row, svd, ...	noise, signal, components, component, filter, signals, source, filters, coefficients, mixture, noisy, sources, ica, separation, mixtures, ...

Fig. 1. DTM and LDA trained on NeurIPS dataset for $K = 60$: Few words of DTM $\tilde{\phi}_0^t$ at $t = 0, 5, 10, 15, 20, 25, 30$ are shown in the first table. Second table shows few words of LDA $\tilde{\phi_{19}}$ and $\tilde{\phi_{55}}$ respectively, and both curves in the graph are the JS similarity measure of $\tilde{\phi}_0^t$ with LDA $\tilde{\phi_{19}}$ and $\tilde{\phi_{55}}$. This is a graphical representation of two fragmented LDA topics related to one single DTM topic.

fragmentation. By subjective analysis, JS value of 0.7 is selected as threshold value for fragmented topics analysis.

Formally, jth DTM topic is *related* to the kth LDA topic if there exists t such that $JSD(\tilde{\phi}_{L,k}, \tilde{\phi}_{D,j}^t) \leq 0.7$. kth and lth LDA topics are fragmented topics of the jth DTM topic if the jth DTM topic is related to both the kth and lth LDA topic.

B. Topic popularity analysis

The time-series topic popularity, which is the second important information offered by DTM, can be extracted from the LDA topics. After calculating document-topic distribution θ_{dk}, the documents are categorized with the same time-series information as used in DTM. Then, we calculate the estimated number of documents for each topic in a time series manner and construct a graph that is comparable to the DTM topics popularity information.

V. Experiment

The experimental process started with collecting and preparing the datasets. Then, appropriate configurations for DTM and LDA models were selected. After training both models with one dataset at a time, we extracted topics-word distributions and word probability. These word probabilities were used for computing the JS divergence and we made the JS similarity-based graphs. We also plotted population graphs using LDA inference to compare it with the DTM topics.

A. Datasets

Three different datasets were used in this experiment.

NeurIPS: This dataset consists of research papers from the conference of neural information processing systems (NeurIPS formally known as NIPS) from 1987 to 2017 (30 time slices).

 ISBN: 978-1-61208-842-6

Total of 7239 research papers were used. In the preprocessing, we removed stop words, special characters, URLs, and words having only two characters because most two characters words do not have concrete meanings.

Twitter: Tweets2011 [21] dataset consists of more than three million English tweets sampled between January 23 to February 8, 2011 (17 time slices). As the original dataset consists of tweets in many languages, we used the Python library *langdetect* to extract the English tweets. Usually, a tweet is a messy piece of text, so some preprocessing is desirable as the first step in cleaning this data. We therefore removed stop words, usernames, URLs, special characters, and two-letter words. After applying day-hashtag pooling, 408,200 documents were obtained and became part of the training dataset.

News: We use Thompson's dataset [22] consists of 204,135 news articles from 18 American publications. There are 191,530 articles that have date information and also the distribution of articles over the years is sparse. We therefore selected articles from 2016 and 2017, totaling 95,997 and 75,034 respectively. Thus, a total of 171,031 news articles were divided into 24 time slices based on the month-year parameter for DTM training and the inference of LDA. The same preprocessing steps were applied to this dataset as mentioned above for the other datasets.

B. Models configuration

LDA with the stochastic variational Bayesian method [23] in Java with number of topics K, 1000 docs per batch, and 1000 iterations was trained with the above-mentioned datasets one at a time.

DTM was implemented using the gensim.model.wrappers with DTM implementation in C and C++. We trained the DTM on three different number of topic configurations with each dataset.

Topics: We selected three values (20, 50, 100) for training time and (30, 60, and 90) for topic drifting and topic popularity as the hyperparameter "number of topics", denoted as K.

VI. Results

This section is divided into multiple sub-sections and each part explains the different aspects of our research.

A. Training time cost

As mentioned earlier, the computational cost for DTM is higher than LDA. However, to determine the difference in training time, we conducted a small sub-experiment in which we trained both models with multiple-size datasets and hyperparameter value K. The dataset used for this experiment was the "Twitter" dataset. Preprocessing cleaning and hashtag pooling were applied before training.

Figure 2, shows that increasing the number of documents or the number of topics increase the training time. For small datasets, the training time of DTM was 10X more than LDA and exceeds **"100 times"** for big datasets. Normally in NLP, datasets are relatively bigger in size, therefore we can say that DTM will take around 100X more time for training compared with LDA under the same conditions.

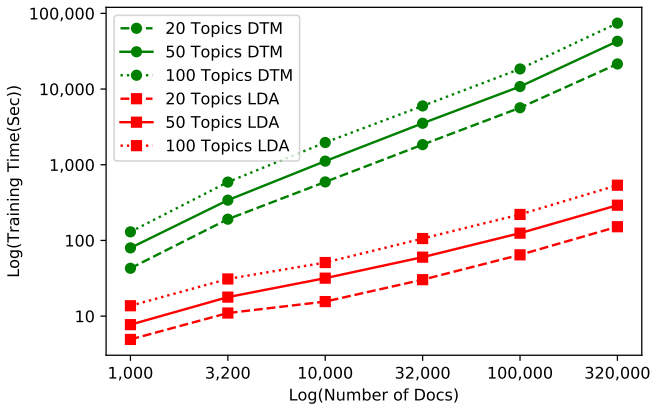


Fig. 2. The graph is in logarithmic scale to fit higher values in the figure. x-axis is document-size value and y-axis is the time in seconds that each model took for training.

B. Topic drifting

A single DTM topic consists of topics at each time slot. For clarity, let us call such a time-slice-topic the "focus" of the DTM topic. The focus of a DTM topic changes over time, as shown in the first part of Figure 1, where the focus changed from "Signal Processing" to "Tensor Decomposition" by the end. This is called topic drifting or topic transition. We calculated the total unique vocabularies for each DTM topic. V_s is the vocabulary size, which is the number of unique words that appeared in all time slots normalized topic-word distributions of a single DTM topic. The minimum vocabulary size for any topic was 50. If any topic had V_s close to this number, it means there were few new words in the different time slot topics. In short, the focus of this specific topic remained the same and there was no topic drifting.

In Table I,$K(V_s > 70)$ means the number of DTM topics having a vocabulary size of more than 70. Similarly $K(V_s > 90)$ and $K(V_s > 120)$ mean the number of topics with V_s more than 90 and 120, respectively. These values for the Twitter dataset are very low, which means there were not many new words in the DTM topics and the focus of the topics remained the same over all times. This implies that topic drifting for the Twitter dataset is negligible. And $K(V_s > v)$ values for the DTM trained on NeurIPS and News dataset were relatively high, which implies that there were topic drifting phenomenons.

C. JS analysis

To extract the information overlap of the DTM and LDA topics, we computed JS values using (5) for all the datasets in all topic configurations. The JS value is bounded by 0 and 1 for two distributions, where 0 means both distributions are identical and 1 means there is no similarity between both probability distributions. A threshold value of 0.7 was selected

 ISBN: 978-1-61208-842-6

TABLE I
TOPIC DRIFTING: $K(V_s > v)$ VALUES INDICATES THE NUMBER OF DTM TOPICS OF WHICH $V_s > v$.
FRAGMENTATION: RT(RELATED TOPIC) AND FT (FRAGMENTED TOPIC) ARE BASED ON JSD VALUES OF LDA TOPICS WITH DTM TOPICS.

Configuration		Topic Drifting			Fragmentation				
Dataset	**Topics**	$K(V_s > 70)$	$K(V_s > 90)$	$K(V_s > 120)$	RT	FT	F 2	F 3	F 4 & more
NeurIPS	30	13	8	0	17	4	3	1	0
	60	58	56	11	42	16	11	5	0
	90	90	90	90	69	28	25	2	1
Twitter	30	3	1	0	5	0	0	0	0
	60	3	0	0	11	1	1	0	0
	90	1	0	0	14	3	2	1	0
News	30	29	20	5	8	1	1	0	0
	60	57	33	1	24	2	2	0	0
	90	83	37	2	42	4	4	0	0

and any DTM topic distribution having a JS value lower than or equal to this threshold when measured with the LDA topic distributions was part of the related topic "RT", fragmented topic "FT", and others. A summary of this analysis is set forth in Table I under Fragmentation column. "RT" is the total number of DTM topics having a relationship with the LDA topics. "FT", "F2", "F 3", and "F 4 & more" are the number of DTM topics having a JS relationship with two or more LDA topics, only two LDA topics, only three LDA topics, and more than three LDA topics, respectively.

Data shows that a negligible amount of "FT" (fragmented topics) was found for the datasets "News" and "Twitter" because most news articles and tweets are instantaneous responses of some events, and these topics die within short period of time; in other words, we see other tweets and article about other events. Due to this focus shifting behavior of the documents, DTM cannot accurately locate topic transitions over time. That is why very few fragmented topics were found for these datasets. Related topics "RT" are comparatively higher for "News" as compared to "Twitter" dataset because the domain of tweets is huge; it could be anything ranging from personal (My pet is very cute) to political (US president announced a restriction on trade agreement with China), whereas the News articles domain is restricted compared with Twitter. We can therefore have many topics in the Twitter dataset and due to random initial conditions of both DTM and LDA, it is safe to say that both models could come up with different topics. As mentioned, the News dataset domain is restricted so we see high topic overlapping in News dataset.

The domain of the "NeurIPS" documents focused on a few subjects (machine learning, artificial intelligence, computational neuroscience, etc), so related topics' "RT" values are very high compared with other dataset configurations. High fragmented topic "FT" values can be seen for the "NeurIPS" dataset in Table I because research papers tend to follow previous researches or somehow align with previous research papers. That is why we can see a well-defined topic transitions in the DTM topics, as shown in Figure 1.

In all the datasets, increasing the number of topics resulted in an increase in "RT" and "FT" values. Table II shows, if we increase the number of topics in LDA, we get more and more fragmented topics, which means that topics are further divided into smaller and more focused themes. DTM's computation cost restricts us from increasing the number of topics, so we cannot get the type of topics that we can get from LDA with a very high K hyperparameter.

TABLE II
FRAGMENTATION BEHAVIOUR WHEN LDA IS TRAINED WITH HIGH NUMBER OF TOPICS: K WAS 30 FOR DTM AND **1000** FOR LDA.

Dataset	**RT**	**FT**	**F 2**	**F 3**	**F 4 & more**
NeurIPS	25	20	3	7	10

D. Time dependent topic popularity

For DTM, time-series topic popularity is estimating the number of documents for each topic at each time slice. We can easily construct this information into a self-explanatory graphical representation of topic popularity. For this analysis, we selected the 60 topics "NeurIPS" configuration. Then, γ distributions for the documents were computed. A γ distribution is the probability of each topic for a document. Summation over the topics then, gives us the document estimation. For LDA, model training is done with same configuration. Inference dataset was constructed before extraction of θ_{dk} distributions by coupling date information with documents.

With θ_{dk}'s and X_t's, by using (1) and (2), we got N_k^t and built graphs. To reduce noise effects and to make the graphs smooth, we used the Savitzky-Golay digital filter [24]. These graphs are shown in Figure 3. These graphs show that the time-series topic popularity can be extracted [14] from DTM as well as LDA (Details in Discussion section).

VII. DISCUSSION

Topic drifting and topic popularity are the main aspects of this research and we compared these aspects for DTM and LDA. In this section, we discuss a few important points of both concepts.

Topic drifting: There is no topic drifting for DTM trained on the "Twitter" dataset (Table I), so the only important information which can be extracted from such datasets is the time-series topic popularity which can be extracted using LDA, thus we should avoid the high cost DTM. For the "NeurIPS" dataset, the topic drifting increased with an increase in number

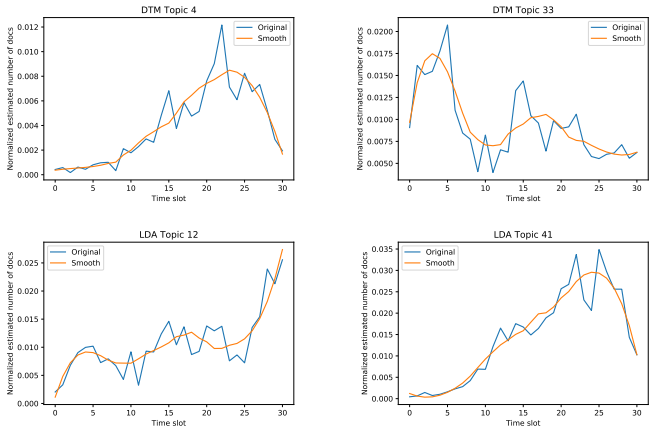


Fig. 3. Two topics from each model are shown here. The horizontal axis shows the time and the vertical axis is the normalized estimated number of documents for these topics.

of topics. All 90 topics have V_s greater than 120 for this dataset (Table I), which means there was high topic drifting which provides rich insights of topic transitions. Therefore, if we want to examine topic drifting in such datasets, DTM is a promising option; however, we must keep in mind that if our goal is topic popularity, then LDA is a far better option. Topic drifting can be experienced for "News" dataset, but the vocabulary size is comparatively low for the higher number of topics. This means that we do have topics drifting with such datasets, but it may not be as effective as we want. More subjective analysis based on problem statement can help to choose better model. An interesting finding is sometimes DTM tends to forcefully find the topic transition. For example, in the 30-topic "News" dataset configuration, topic 29 started with words *(archive, team, collection, sign, projects, machine, contains, lost, websites, wayback)*, but the word distribution at the end was *(travel, airport, flight, trip, passengers, travelers, plane, airlines, united, airline)*. Looking at these distributions, we can say that DTM failed to extract the correct topic drifting for topic 29.

Topic popularity: Once the models are trained, we can extract γ and θ distributions for any document. With θ distributions using the method described in Section 3 for the LDA model, we can construct topic popularity graphs. Similarly, we can construct these graphs for DTM topics using γ distributions. Thus, this information can be extracted using both models. Notably, the topic popularity extracted from DTM is a little vague because DTM topics have topic transition information embedded within the topics. To explain this phenomenon with an example, we manually selected 2 topics from DTM and 2 topics from LDA; DTM topic 4 shown in Figure 3 is *(retrieval, content, query, text, semantic, lda, relevant, word, latent, topics)* at T_0, which is about "Information retrieval from documents" and the word distribution at T_{30} is *(topic, document, lda, word, topics, latent, dirichlet, text, allocation, model)*, which is about "Document analysis with LDA". Similarly, DTM topic 33 was about "Language structure rules" at initial time slots and the theme of the topic was changed to "Question-answer reasoning" around at the end. Therefore, if we are looking for the popularity graph of a topic "Information Retrieval", then LDA topic 12 is a more accurate option. Similarly, LDA topic 41 is more accurate if we want to see the popularity graph of the topic "Variational topic model LDA" because there is no topic drifting in LDA. The word distribution for LDA topic 12 is *(word, language, sequence, recurrent, text, semantic, context, attention, table, embedding)* and for LDA 41 is *(latent, topic, sampling, mixture, gibbs, dirichlet, lda, markov, document, likelihood)*. Because of the topic transition information embedded with DTM topics, DTM is not the best option for time-series topic popularity information.

VIII. Conclusion

In this research, we executed a comprehensive study on the time-series analysis of the popular topic models DTM and LDA. Our research focused on the time-series information of topic drifting and topic popularity. To compare both models, we tried to extract this information from the topic distributions. Multiple datasets along with multiple topic configurations were used for this experiment.
Our findings are:

1) DTM takes 100 times longer to train the model as compared to LDA for large datasets.
2) Topic drifting is a unique property of DTM that is difficult to extract from LDA, but datasets like "Twitter" do not have topic transition information, so applying DTM to such datasets is waste of resources.
3) Time-series topic popularity can be extracted from both models, but it is precise from LDA because DTM has topic transition information embedded in the topics.

Fragmentation of topics was also detected in this process from the datasets focused on one domain, e.g., "NeurIPS", which is another interesting aspect of this research and could be studied in the future. To summarize, topic popularity — common information needed as time series information — should be extracted using LDA because it is faster and provides concrete information. However, if topic drifting is required, then DTM is the only option, although sometimes, it may give inaccurate information.

Based on our findings and research experiment with multiple datasets configurations, we suggest the usage of DTM and LDA in different case scenarios (Table III).

TABLE III
SUGGESTIONS BASED ON FINDINGS

Use LDA for	**Use DTM for**
Twitter with high K	NIPS data with few number of topics
News with high K	News with smaller K
Short duration datasets e.g. Twitter	Long duration docs e.g. NIPS
Extract topic popularity	Extract topic drifting

 ISBN: 978-1-61208-842-6

REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, vol. 3, pp. 993-1022, 2003

[2] D. M. Blei and J.D. Lafferty, "Dynamic topic models," Proceedings of the 23rd international conference on Machine learning, 2006, pp.113-120

[3] D. Koike, Y. Takahashi, T. Utsuro, M. Yoshioka, and N. Kando, "Time series topic modeling and bursty topic detection of correlated news and twitter," Proceedings of the Sixth International Joint Conference on Natural Language Processing, 2013, pp.917-921

[4] Noriaki Kawamae, "Trend analysis model: trend consists of temporal words, topics, and timestamps," Proceedings of the fourth ACM international conference on Web search and data mining, 2011, pp. 317-326

[5] H. Zhang, G. Kim, and P. E Xing, "Dynamic Topic Modeling for Monitoring Market Competition from Online Text and Image Data," Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp.1425-1434

[6] J. Kalyanam, A. Mantrach, D. Saez-Trumper, H. Vahabi, and G. Lanckriet, "Leveraging Social Context for Modeling Topic Evolution," Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp.517-526

[7] W. Xie, F. Zhu, J. Jiang, E. Lim, and K. Wang, "TopicSketch: Real-Time Bursty Topic Detection from Twitter," IEEE Transactions on Knowledge and Data Engineering, vol. 28, pp.2216-2229, 2016

[8] H. Amoualian, M. Clausel, E. Gaussier, and M. R. Amini, "Streaming-LDA: A Copula-based Approach to Modeling Topic Dependencies in Document Streams," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp.695-704

[9] Y. Kim, J. J. Han, and C. Yuan, "TOPTRAC: Topical Trajectory Pattern Mining," Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp.587-596

[10] A. Acharya, J. Ghosh, Jand M. Zhou, "A Dual Markov Chain Topic Model for Dynamic Environments," Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp.1099-1108

[11] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, "Reducing the sampling complexity of topic models," Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 891-900

[12] L. Yut, C. Zhang, Y. Shao, and B. Cui, "LDA* a robust and large-scale topic modeling system," Proceedings of the VLDB Endowment, 2017, pp.1406-1417

[13] J. Chen, J. Zhu, J. Lu, and S. Liu, "Scalable training of hierarchical topic models," Proceedings of the VLDB Endowment, 2018, pp.826-839

[14] M. H. U. R. Khan, K. Wakabayashi, and S. Fukuyama, "Events Insights Extraction from Twitter Using LDA and Day-Hashtag Pooling," Proceedings of the 21st International Conference on Information Integration and Web-based Applications Services, 2019, pp.240-244

[15] M. Huang, M. Zolnoori, J. E. Balls-Berry, T. A. Brockman, C. A. Patten, and L. Yao, "Technological innovations in disease management: text mining US patent data from 1995 to 2017," Journal of medical Internet research, vol. 21, 2019

[16] B. Han, P. Cook, and T. Baldwin, "Automatically constructing a normalisation dictionary for microblogs," Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, 2012, pp. 421-432

[17] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter," Proceedings of the first workshop on social media analytics, 2010, pp.80-88

[18] W. X. Zhao, J. Jiang, J. Weng, J. He, E. P. Lim, H. Yan, and X. Li, "Comparing twitter and traditional media using topic models," European conference on information retrieval, 2011, pp. 338-349

[19] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie, "Improving lda topic models for microblogs via tweet pooling and automatic labeling," Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, 2013, pp. 889-892

[20] J. Lin, "Divergence measures based on the Shannon entropy," IEEE Transactions on Information theory, vol. 37, pp. 145-151, 1991

[21] TREC, "Tweets2011," https://trec.nist.gov/data/tweets/, 2011, [Online; accessed 2021.03.15]

[22] A. Thompson, "204,135 articles from 18 American publications," https://components.one/datasets/all-the-news-articles-dataset/, 2018, [Online; accessed 2021.03.15]

[23] D. Mimno, M. Hoffman, and D. Blei, "Sparse stochastic inference for latent Dirichlet allocation," Proceedings of the 29th International Coference on International Conference on Machine Learning, 2012, pp. 1515–1522

[24] H. William and S. A. Teukolsky, "Savitzky-Golay smoothing filters," Computers in Physics, vol. 4.6, pp. 669-672, 1990

 ISBN: 978-1-61208-842-6